NPS-55-89-02

NAVAL POSTGRADUATE SCHOOL

Monterey, California

CONSTANT ACCESS SYSTEMS: A GENERAL FRAMEWORK FOR GREEDY OPTIMIZATION ON STOCHASTIC NETWORKS

MICHAEL P. BAILEY

MARCH 1989

Approved for public release; distribution is unlimited.

Prepared for:
Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL,
MONTEREY, CALIFORNIA

Rear Admiral R. C. Austin
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research conducted under the Naval Postgraduate School Research Council Program.

Reproduction of all or part of this report is authorized.

This report was prepared by:

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): NPS55-89-02	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): 55	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION: Naval Postgraduate School; 8b OFFICE SYMBOL (If applicable): 55Ba	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: O&MN, Direct Funding
8c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) Constant Access Systems: A General Framework for Greedy Optimization on Stochastic Networks.

12 PERSONAL AUTHOR(S) Michael P. Bailey

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical	FROM ____ TO ____	March 1989	38

16 SUPPLEMENTARY NOTATION
The views expressed in this paper are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Stochastic networks, stochastic optimization

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

We consider network optimization problems in which the weights of the edges are random variables. We develop conditions on the combinatorial structure of the problem which guarantee that the objective function value is a first passage time in an appropriately constructed Markov process. The arc weights must be exponentially distributed, the method of solution of the deterministic problem must be greedy in a general sense, and the accumulation of objective function value during the greedy procedure must occur at a constant rate. We call these structures constant access systems after the third property. Examples of constant access systems include the shortest path system, time until disconnection in a network of failing components, and some bottleneck optimization problems. For each system, we give the distribution of the objective function, the distribution of the solution of the problem, and the probability that a given arc is a member of the optimal solution. We also provide easily implementable formulae for the moments of these quantities.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL: Michael P. Bailey	22b TELEPHONE (Include Area Code): (408) 646-2085; 22c OFFICE SYMBOL: 55Ba

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986-606-243

UNCLASSIFIED

CONSTANT ACCESS SYSTEMS: A GENERAL FRAMEWORK FOR GREEDY OPTIMIZATION ON STOCHASTIC NETWORKS

by

Michael Page Bailey

Department of Operations Research
U. S. Naval Postgraduate School
Monterey, California

ABSTRACT

We consider network optimization problems in which the weights of the edges are random variables. We develop conditions on the combinatorial structure of the problem which guarantee that the objective function value is a first passage time in an appropriately constructed Markov process. The arc weights must be exponentially distributed, the method of solution of the deterministic problem must be greedy in a general sense, and the accumulation of objective function value during the greedy procedure must occur at a constant rate. We call these structures constant access systems after the third property. Examples of constant access systems include the shortest path system, the longest path system, time until disconnection in a network of failing components, and some bottleneck optimization problems. For each system, we give the distribution of the objective function, the distribution of the solution of the problem, and the probability that a given arc is a member of the optimal solution. We also provide easily implementable formulae for the moments of these quantities.

1.0 INTRODUCTION

In this paper, we unify a set of results concerning the performance of networks with random arc weights. Finding the distributions of the shortest path, longest path, and maximum flow were provided in Kulkarni[1987], Kulkarni and Adlakha[1986], and Kulkarni [1987], where each problem was considered on a network with independent, exponentially distributed arc weights. In each case, a Markov process was constructed for which the first passage time to a set of states was the optimal objective function value of the problem. Thus, the distribution, moments, etc., of the optimal objective function value could be found using standard Markov process technique.

The same methodology was applied to the nonplanar maximum flow problem and to Prim's spanning tree problem, neither attempt being successful. We could not construct a Markov process with first passage time equal to the objective function value for either problem, though these problems seem closely related to the problems for which the method was successful. This left the investigators with the problem of determining which characteristics of these optimization problems was essential in order to ensure that a Markov process solution exists.

This paper provides a general framework for randomly weighted network optimization problems which have optimal objective function values are given by absorption times of some Markov process. The restrictions which describe this class are that the method of solution is essentially greedy, and that the underlying combinatorial structure have two properties which we call the constant access and interval properties. For any combinatorial optimization problem with these properties, we give the construction of the Markov process required. For each system, we give the distribution of the objective function, the distribution of the solution of the problem, and the probability that a given arc is a member of the optimal solution. We also provide easily implementable formulae for the moments of these quantities. All of these calculations exploit the uppertriangularity of the generator matrix of the constructed Markov process.

Awareness of a general class of constant access systems allows us to consider other combinatorial optimization problems and to determine if they may be extended to the randomly weighted case using Markov processes. By pinpointing the required properties for successful application of Markov processes to the objective function distribution problem, we can narrow the search for problems for which we expect success. Stated differently, we have shown that the mechanism of transition in Markov processes is identical to the mechanism of access of the greedy algorithm.

Let E be a set of elements, typically the edges of a graph, and let B be a set of strings made up of elements in E. For each $Y \in B$, let $W(Y)$ be a random variable corresponding to the objective function value of Y. We are interested in computing

$$F(Y, t) = P[W(Y) \leq t,\ W(Y) \leq W(Y') \text{ for all } Y' \in \mathbf{B}],$$

the probability that Y is the minimum weight element of **B** and its value does not exceed t. The marginals of F may be used to find the distribution of the optimal objective function value, criticality indices for each arc, and other interesting network performance measures. In Dijkstra's shortest path algorithm (forward looking version), **B** is the set of states which have the destination permanently labeled. If Y is one such state, $F(Y, t)$ is the probability that Y is the terminating state of the algorithm and the shortest path has length not exceeding t. If we aggregate all of the states which give the same shortest path, we can sum the probabilities to find the probability that a given path is shortest and its length does not exceed t, and we can let $t \to \infty$ to find the probability that a given path is shortest.

In this work, we will identify a class of problems for which a stochastic process $\{X(t), t \geq 0\}$ may be constructed such that each $Y \in \mathbf{B}$ is an absorbing state of $\{X(t), t \geq 0\}$. The fundamental property of $\{X(t)\}$ is that $F(Y, t)$ is the distribution of the time of $\{X(t)\}$'s first passage to Y. Characteristics of this class of combinatorial problems include optimality of the greedy algorithm for the underlying deterministic combinatorial problem, as well as two properties called the interval and constant access properties. The stochastic process $\{X(t), t \geq 0\}$ is shown to be a generalized semi-Markov process (GSMP). When the arc weights are exponentially distributed and independent, $\{X(t), t \geq 0\}$ is a Markov process.

In Section 2 we outline the combinatorial structures which we will treat. Section 3 contains the combinatorial results required to pursue the case of random weights. Section 4 contains the analysis of stochastic combinatorial problems, including the conditions under which the stochastic system is a Markov process. We use the Markov process to derive distributions of interest and we include some results on computational efficiency. In Section 5 we detail some important examples of constant access systems. While some of these systems have been investigated ad hoc in [1987], [1986], and [1987], several of these systems are new to the literature. They were uncovered as constant access systems as a result of the construction of the general system. Section 6 contains our conclusions and some comments concerning extensions of this work.

2.0 BASIC COMBINATORICS

In this section, we establish the terminology and notation necessary to describe the systems we study. We define the minimizing greedy algorithm and give Dijkstra's shortest path system as an example.

Let E be a set which we refer to as the **ground set**, and let ζ be a set of simple strings (ordered sets with no repeats) of elements in E. Let $X \in \zeta$, then the i-th element in the string X is denoted x_i, and the length of the string is given by $|X|$. We will denote the operation of concatenation as "$\cdot$". Thus if $X = (x_1, x_2, \ldots, x_k)$, $\alpha \in E$, and $Y = (y_1, y_2, \ldots, y_m)$, then $X\cdot\alpha = (x_1, x_2, \ldots, x_k, \alpha)$, and $X\cdot Y = (x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_m)$. For the string $X = (x_1, x_2, \ldots, x_k)$, the string made up of the first i elements, $i \le k$, will be denoted X_j, $X_i = (x_1, x_2, \ldots, x_i)$, and is called a **prefix** of X. We will use set operations $\cap$, $\cup$, $\subset$, and $\in$ on the strings of ζ to indicate the operation performed on the underlying set of the string.

In this work, we will deal with string systems have the following properties:

i) $\phi \in \zeta$;

ii) $X \in \zeta$ implies that $Y \in \zeta$ for all $Y \subset X$.

The set ζ is called the set of **feasible strings**.

We often wish to refer to the set of elements of E which we can feasibly append to the string $X \in \zeta$, denoted as A(X). This set is called the set of **accessible elements of X** and is formally given by $A(X) = \{x \in E\colon X\cdot x \in \zeta\}$. When an element of A(X) is appended to X, it is said to have been **accessed**. The set B is defined as $B = \{X \in \zeta\colon A(X) = \varnothing\}$, and is called the set of **basic** strings, following the terminology used in matroid theory. An **access chain** to $X \in \zeta$, $|X| = k$, is the sequence of feasible strings $X_0 = \phi, X_1, X_2, \ldots, X_k = X$. Note that for any $X \in \zeta$, $|X| = k$, $\{Y \in B\colon X \text{ is on a chain of access to } Y\} = \{Y \in B\colon Y_k = X_k\}$.

2.1 Clutter Intersection Systems

Let $v\colon E \to \Re^+$ be a nonnegative weight function on the set E. For each $Y \in B$, define $d^*(Y)$ to be a known subset of $\{Y\}$ called the **determining structure** of Y. Let w be the objective function on B given by

$$w(Y) = \Sigma_{x \in d^*(Y)}\, v(x),$$

thus, each basic element's weight is linear in its determining structure. Note that if $d^*(Y)$ equals the underlying set of Y, then w is the linear objective function common to matroid analysis.

In our development, we will make the following two assumptions concerning determining structures of basic elements:

i) $\bigcup_{Y \in B} d^*(Y) = E$

ii) $d^*(Y) \not\subset d^*(Z)$ for any $Y, Z \in B$, $Y \neq Z$.

Thus the set $\{d^*(Y): Y \in B\}$ is a **clutter** on the set E, see Edmonds and Fulkerson [1970]. Assumption i guarantees that every member of E belongs to at least one determining structure. Assumption ii guarantees that for each determining structure of a basic element, there exists a weight function for which that determining structure is of minimum weight. In a two terminal undirected network in which each arc is on a path between the terminals, the set of paths is a clutter on the set of arcs in a connected network, as is the set of (minimal) cutsets.

The set $\{d^*(Y): Y \in B\}$ fully describes the objective function w for basic strings of ζ, however w(X) for $X \in \zeta - B$ is still left undetermined. We extend the notion of determining structures to nonbasic strings. Let X be a nonbasic element of ζ. Define $\Gamma(X)$ as $\Gamma(X) = \{d^*(Y): X \subset Y, Y \in B\}$. Thus, $\Gamma(X)$ is the set of determining structures of basic strings which can terminate any chain of access containing X. Let

$$w(X) = \max_{d^*(Y) \in \Gamma(X)} \Sigma_{d^*(Y) \cap X}\ v(x) \qquad (2.1)$$

and let $d^*(X) = d^*(Y) \cap X$ for the maximizing $Y \in \Gamma(X)$. Note that $d^*(X)$ implicitly depends on the weight function v. The triple (E, ζ, d^*) will denote a system with ground set E, feasible strings z, and determining structure function d^*.

Example 1. (A Shortest Path Example)

Consider the graph G = (N, E) in Figure 1.

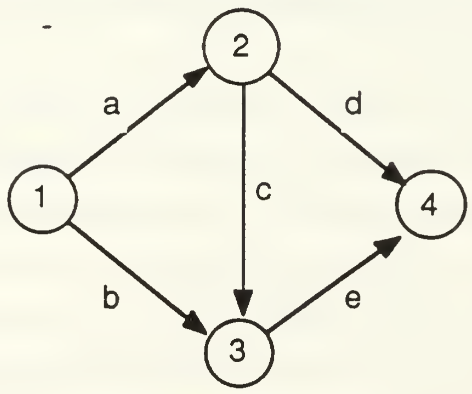

Figure 1.

Let E be the set of arcs. Construct ζ so that $X \in \zeta$ if X is the arc set of a directed tree rooted at node 1. Further restrict ζ by requiring that any string containing an arc incident with node 4 is basic.

$E = \{a, b, c, d, e\}$, and $\zeta = \{\phi$, a, b, ab, ac, ad, ba, be, abd, abe, acd, ace, bad, bae$\}$, $B = \{$ad, be, abd, abe, acd, ace, bad, bae$\}$. Note that adb $\notin \zeta$ because ad an arc incident with node 4, thus $A(ad) = \emptyset$. For each $Y \in B$, let us define the determining structure of Y as the (1, 4) directed path contained in Y. Thus,

$d^*(ad) = \{a, d\}$	$d^*(acd) = \{a, d\}$
$d^*(be) = \{b, e\}$	$d^*(ace) = \{a, c, e\}$
$d^*(abd) = \{a, d\}$	$d^*(bad) = \{a, d\}$
$d^*(abe) = \{b, e\}$	$d^*(bae) = \{b, e\}$.

Note that each element of E is contained in at least one determining structure. Also note that no determining structure is strictly included in another. Thus $\{d^*(Y): Y \in \mathbf{B}\}$ is indeed a clutter on E. Note also that several elements of z have the same determining structure, thus their objective function values are equal. Without knowledge of v, we cannot say what the determining structure of nonbasic strings are. For instance, the determining structure of the string ab is the longer of the arcs a and b.

2.2 Greedy Minimizing Algorithm

Let $X \in \zeta$. Define the greedy generator G as the function $G(X) = \operatorname{argmin}_{x \in A(X)} w(X \cdot x)$. A greedy chain of access is a chain of access $X_0 = \emptyset, X_1, X_2, ..., X_k$ such that $X_{i+1} = X_i \cdot G(X_i)$. The greedily accessed basic string will be denoted with a subscript "G". A greedy system has the property that for any nonnegative weight function v, $w(Y_G) \leq w(Y)$ for all $Y \in \mathbf{B}$.

Example 2. (The Greedy System)

Recall our network example for Figure 1. Let the weight function v be given by the following table.

x:	a	b	c	d	e
v(x):	2	4	5	4	1

Let $OUT(n) = \{x \in E: tail(x) = n\}$ and $IN(n) = \{x \in E: head(x) = n\}$ for each $n \in N$. The following algorithm is the greedy algorithm for this system

initialize:

$X_0 = \emptyset$

$A(X_0) = OUT(s)$

$w(X_0) = 0$

for each $x \in E$, $r(x) = v(x)$

$i = 0$

While $|X_i \cap IN(t)| = \emptyset$

$x_{i+1} = argmin_{x \in A(X_i)}\ r(x)$

$X_{i+1} = X_i \cdot x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

$A(X_{i+1}) = A(X_i) - \{x_{i+1}\} - \{x \in E$: *no* $(tail(x), t)$ *directed path exists in* $(N, E - X_{i+1}) \cup OUT(head(x_{i+1}))\}$

$i = i + 1$

endwhile

The set of feasible strings z is the set of strings which this algorithm generates for all possible nonnegative length functions v. The following table gives the outcome of the algorithm for the network above.

i	X_i	$d^*(X_i)$	$w(X_i)$	$x \in A(X_i)$	$w(X_i \cdot x)$
0	∅	∅	0	a	2
				b	4
1	a	{a}	2	b	4
				c	7
				d	6
2	ab	{b}	4	d	6
				e	5
3	abe	{b, e}	5	–	–

The example ends with $X_3 = X_G =$ abe. •

The reader should recognize this algorithm as Dijkstra's shortest path algorithm. In fact, several of our greedy systems are recasts of well-known network optimization schemes, the hereditary family (E, ζ) being set of sample paths for the algorithm. While this representation is inefficient for the deterministic problem, it is vital for the extension to the case of random weights.

3.0 CONSTANT ACCESS AND INTERVAL PROPERTIES

We must further restrict our optimization scheme in order to guarantee memorylessness of the random weighted extension. Ideally, further restriction of the deterministic system should be independent of v except where it impacts

d^*, and locally verifiable, having to do with the relationship of X to A(X). The following two properties meet this ideal and guarantee memorylessness to randomly weighted systems.

3.1 Interval Property

The following property, the interval property, is an extension of the interval property given by Korte' and Lovasz [1984] in their description of a set of greedoids called alternative precedence structure (APS) greedoids.

Let (E, ζ) be a hereditary language. Let X, Y $\in \zeta$ such that $X \subset Y$. (E, ζ) has the **interval property** if for all Z $\in \zeta$ with $X \subset Z \subset Y$, we have

$$[A(X) \cap A(Y)] \subset A(Z).$$

Thus, during construction of a chain of access, if $x \in E$ becomes accessible at some stage, call it j(x). x remains accessible until either it is accessed or removed from the set of accessible elements. Once an element is removed from the set of accessible elements, it will never be accessible again.

Example 3.

Returning to the ongoing example, consider arc d. At the onset, d is not accessible. Consider the string abe $\in$ **B**. For this string, d is accessible once a has been accessed, thus j(d) = 1. d remains accessible during stages 2 and 3. Once e is appended to ab, d is no longer accessible. This property can be verified for all elements of E for each member of **B**. •

3.2 Constant Access Property

The system (E, ζ, d^*) has the **constant access property** if for each X $\in \zeta$ – **B** and each Y $\in$ **B** with $X \subset$ Y,

$$|A(X) \cap d^*(Y)| = 1.$$

Equivalently, (E, ζ, d^*) has the constant access property if for all $x \in A(X)$, $x = X \cap P$ for some $P \in \Gamma(X)$. In a subsequent investigation, we will generalize this property to allow $|A(X) \cap d^*(Y)| = n(X)$, where n(X) is a known constant depending only on X.

Example 4.

In our ongoing example, note that for each $X \in \zeta - \mathbf{B}$, $A(X)$ is contained in a (1, 4) cutset and consists of the arcs pointing toward node 4 (a uniformly directed cutset, UDC), resulting in $|A(X) \cap P| = 1$ for all sets P which are (1, 4) directed paths, (Sigal, et. al. [1980]). •

Greedy systems which have the constant access and interval properties, are greedy systems, and have determining structures which form a clutter on the ground set are called **constant access systems.**

3.3 Systems with the Constant Access Property

Henceforth, (E, ζ, d^*) will be assumed to be a constant access system. We will now present a set of results which give a formula for the cost of constructing the greedy chain of access. This formula is the key to the proof of memorylessness in the next section, as well as enhancing our understanding of deterministic greedy access systems.

For any $X \in \zeta$, $X = (x_1, x_2, ..., x_n)$, let $j(x_i) = \min\{j: x_i \in A(X_j)\}$ for each $i = 1, 2, ..., n$. Thus, $j(x_i)$ is the first stage for which x_i is accessible.

Lemma 1. Let $Y \in \mathbf{B}$, $Y = (y_1, y_2, ..., y_m)$. Let i and k be such that $i \le k \le m$. Then $y_i \in d^*(Y_k)$ implies that $y_{j(y_i)} \in d^*(Y_k)$.

Proof. Consider the stage $j(yi) - 1$, the stage immediately preceding the stage when y_i becomes accessible. By the constant access property, there exists unique $x \in d^*(Y_k)$ in the set $A(Y_{j(y_i)-1})$. If $y_i \in A(Y_{j(y_i)}) \cap d^*(Y_k)$, then $x \in A(Y_{j(y_i)})$ because only one member of $d^*(Y_k)$ may be in $A(Y_{j(y_i)})$. Thus x is in $A(Y_{j(y_i)-1})$, x is not in $A(Y_{j(y_i)})$, and $x \in d^*(Y_k)$ hence $x \in Y_k$. The interval property guarantees that $x = yj_{(yi)}$. •

Upon reflection, lemma 1 gives good insight into the workings of access systems with the constant access and interval properties. It states that once one member of the determining structure is identified, the members of the determining structure accessed before it are known automatically. In the next lemma, we establish that if an element X is generated greedily, the last element of X is a member of d(X).

Lemma 2. Let (E, ζ, d^*) be a constant access system. Let X be a greedily generated string, $|X| = n$. Then for each $x \in A(X)$, $x \in d^*(X \cdot x)$.

Proof. See appendix A.

Lemma 2 is based on some observations about the greedy generator. Consider the possibility that $x \notin d^*(X \cdot x)$. In this case $w(X) = w(X \cdot x)$ and accessing x is "free." Two possibilities exist. Possibly x has been accessible before stage m, in which case we would have accessed this low cost element before this stage. The other possibility is that x was not accessible before stage m. In this case, $j(x) = m$. This implies that x and x_m are both members of some element K of $\Gamma(X \cdot x)$ and

$$\Sigma_{z \in K \cap (X \cdot x)} v(z) = v(x) + \Sigma_{z \in K \cap X} v(z)$$

so accessing x is certainly not costless. Hence we derive a contradiction in either case.

Thus, the greedily generated elements $X \in \zeta$ are closed in the sense that feasibly adding any element of $E - X$ increases the objective function value of the string.

Lemma 3. Let $X_i \in \zeta$ be generated by the greedy generator. Let $\tau_k = w(X_k) - w(X_{k-1})$, $k = 1, 2, \ldots, i$. For each $z \in A(X_i)$, let $C_z = v(z) - [\tau_{j(z)+1} + \tau_{j(z)+2} + \ldots + \tau_i]$. Then

$$w(X_i \cdot z) - w(X_i) = C_z. \tag{3.1}$$

Proof. Consider $z \in A(X_i)$. Let $J \in \Gamma(X_i \cdot z)$ such that

$$\Sigma_{x \in J \cap (X_i \cup \{z\})} v(x) \geq \Sigma_{x \in J' \cap (X_i \cup \{z\})} v(x)$$

for all $J' \in \Gamma(X_i \cdot z)$, thus $J \cap (X_i \cup \{z\}) = d^*(X_i \cdot z)$. By lemma 2, we know that $z \in J$, and by the constant access property $\{x_{j(z)+1} + x_{j(z)+2} + \ldots + x_i\} \cap d^*(X \cdot z) = \varnothing$. Thus $w(X_i \cdot z) = w(X_{j(z)}) + v(z)$. Thus, the definition of the τ's gives us the result

$$w(X_i \cdot z) - w(X_i) = v(z) - [\tau_{j(z)+1} + \tau_{j(z)+2} + \ldots + \tau_i]. \quad \bullet \tag{3.2}$$

Corollary 4. $G(X_i) = \text{argmin}_{z \in A(X_i)} C_z$.

Proof. A result of lemma 3 and the definition of the greedy algorithm. •

Lemma 3 enables us to directly compare the incremental costs of the elements appendable at a given stage. In the sequel, we will consider the case where $\{v(x): x \in E\}$ is a set of independent, exponentially distributed random variables. We will use lemma 3 to show that at each stage, the costs C_z are memoryless with respect to the elements already accessed, and thus the incremental costs of access remain independent and exponentially distributed.

Example 5.

Let us return again to our ongoing example, the weights of the arcs given by

x:	a	b	c	d	e
v(x):	2	4	1	4	1

Consider the string $X = ac$, which the greedily generated string of length 2. $A(ac) = \{d, e\}$, with $j(d) = 1$ and $j(e) = 2$. By lemma 2, $d \in d^*(acd)$ and $e \in d(ace)$. By lemma 1 we identify $d^*(acd)$ as $\{d, x_{j(d)}, \ldots\} = \{d, a\}$, and $d^*(ace) = \{e, x_{j(e)}, \ldots\} = \{e, c, a\}$, both sets are arc sets of (1, 4) directed paths. By lemma 3, $w(acd) - w(ac) = C_d = v(d) - \tau_2 = v(d) - [w(ac) - w(a)] = v(d) - [v(c) - 0] = 3$. $w(ace) - w(ac) = C_e = v(e) - 0 = 1$.

Y_G = ace, as predicted by corollary 4. •

4.0 STOCHASTIC CONSTANT ACCESS SYSTEMS

We are primarily interested in models of system behavior in which the critical measure of performance is a sum of the weights of a determining structure of a basic element. In the last section, we have developed characteristics of these types of optimization problems when the weight function is a known function $v{:}E \rightarrow \Re^+$. We now wish to consider situations in which $\{V(x) : x \in E\}$ is a set of random weights of ground set elements, however, we wish to solve the minimum weight basic element problem using greedy minimization which implies definite knowledge of the incremental costs of adding various elements of the accessible set at each stage. We thus propose the following scenario: at the time that the optimization is to take place, the ground set weights (arc lengths in our examples) are known. However, prior to any realization of the problem being encountered, $\{V(x) : x \in E\}$ is a set of random weights for which we have distributional knowledge. Thus, from a **strategic** point of view, the constant access system being considered may be analyzed as a system with stochastic performance. **Tactically**, the system is seen as a deterministic greedy problem.

Accordingly, we will now consider the case of optimization in the context of uncertainty. Let $\{V(x) : x \in E\}$ be a set of nonnegative random weights, and let $\{W(X) : X \in \zeta\}$ be the associated set of stochastic determining structure objective function values. Thus, we wish to derive the joint distribution function F given by

$$F(Y, t) = P[W(Y) \leq t, W(Y) \leq W(Y') \text{ for all } Y' \in \mathbf{B}]$$

for each $Y \in \mathbf{B}$. With F in hand, the derivation of

$$p(Y) = \lim_{t \rightarrow \infty} F(Y, t);$$

$$F(t) = \Sigma_{Y \in \mathbf{B}} F(Y, t);$$

which represent the probability a basic element Y is optimal and the distribution of the minimum weight basic element, resp., are direct. We will also demonstrate some straightforward methods for deriving performance measure distributions, expected values, and conditional expected values, all of which exploit the properties of the Markov process we develop.

4.1 Sample Paths of the Greedy Minimizing Algorithm

As mentioned in the introduction, we will model the execution of the greedy algorithm as a stochastic process, a novelty in the literature of stochastic combinatorial optimization. Let $\{X(t), t \geq 0\}$ be a time homogeneous stochastic process on state space ζ with transition epoches $S_0 = 0, S_1, S_2, \ldots$, and intertransition times $\tau_i = S_i - S_{i-1}$. $\{X(t), t \geq 0\}$ has the following qualities:

i) $P[X(0) = \varnothing] = 1$;

ii) $P[X(t + s) = Y \mid X(s) = Y] = 1$ for all $Y \in \mathbf{B}$, $s,t \geq 0$;

iii) $P[X(S_n) = X \cdot x, \tau_n = t \mid X(S_{n-1}) = X]$

$$= P[W(X{\cdot}x) - W(X) \leq W(X{\cdot}y) - W(X) \text{ for all } y \in A(X), W(X{\cdot}x) - W(X) = t].$$

Hence, $\{X(t), t \geq 0\}$ starts in $\varnothing$, and is absorbed in any element of $\mathbf{B}$. Furthermore, the probability of making transition from X to $X \cdot x$ occurs after $W(X \cdot x) - W(X)$ time units, and then only if transition to $X \cdot y$ for some other $y \in A(X)$ hasn't already occurred. By virtue of i – iii, we may extend the definition of F to include nonbasic elements of ζ:

$$F(X, t) = P[W(X) \leq W(X') \text{ for } \underline{\text{all}}\ X \in \zeta, |X'| \geq |X|, W(X) \leq t].$$

Thus, $F(X, t)$ is the probability that X is on the (random) greedy sequence and that its objective function value is less than or equal to t. Since the length of X increases at each stage, we may guarantee that the transition matrix of the underlying discrete process of $X(t)$ is uppertriangular by assuming that the elements are listed in increasing length.

Note that without any further distributional assumptions, $\{X(t), t \geq 0\}$ is a generalized semi-Markov process (GSMP), and that imposition of the interval property on (E, ζ) restricts $\{X(t), t \geq 0\}$ to the set of noninteruptive GSMPs, see Schassberger [1976]. However, results concerning GSMPs are almost exclusively concerned with steady state behavior of the system. Because the characteristics of $\{X(t), t \geq 0\}$ we seek concern transient behavior, GSMP theory is of little help in the current context. Research into characteristics of GSMPs with uppertriangular transition matrices may be motivated by the systems discussed in this paper.

4.2 Exponentially Distributed Weights

Let $\{V(x) : x \in \bar{E}\}$ be a set of mutually independent, exponentially distributed random variables with rates $\{\mu(x) : x \in E\}$. Recall that $j(y) = \min \{j: y \in A(Y_j)\}$.

Lemma 5. Let $Y \in \zeta$, $|Y| = n$. Let $y \in A(Y)$. Then

$$P[V(y) > t + \tau_{j(y)+1} + \tau_{j(y)+2} + \dots + \tau_n \mid X(S_i) = Y_i, \tau_{j(y)+1}, \tau_{j(y)+2}, \dots, \tau_n]$$
$$= P[V(y) > t] = e^{-\mu(y)t}. \tag{4.1}$$

Furthermore, given $X(S_i) = Y_i, \tau_{j(y)+1}, \tau_{j(y)+2}, \dots, \tau_i$, the set of random variables $\{V(y) - \tau_{j(y)+1} - \tau_{j(y)+2} - \dots - \tau_i : y \in A(Y_i)\}$ is a set of mutually independent random variables.

Proof. We induct on i. For $i = 0$, the proposition simplifies to the assumption of independent, exponentially distributed weights, $j(x) = 0$ for all $x \in A(\emptyset)$.

Consider the lemma statement as an induction hypothesis true for 0, 1, ..., i. For <u>every</u> $y \in A(Y_i)$, equation 4.1 is true for $j(y) + 1, \dots, i$. Note that the condition $X(S_i) = Y$ implies that for every $y \in A(Y)$, $V(y) > \tau_{j(y)+1} - \tau_{j(y)+2} - \dots - \tau_i$. Thus

$$\tau_i = \min \{V(y) - \tau_{j(y)+1} - \tau_{(y)+2} - \dots - \tau_n : y \in A(Y_i)\} \sim \min \{V(y) : y \in A(Y_i)\}$$

is the minimum of a set of mutually independent, exponentially distributed random variables, which implies that $\tau_{i+1} \sim \exp \bar{\mu}(Y_i)$, where

$$\bar{\mu}(Y_i) = \Sigma_{y \in A(Y_i)} \mu(y). \tag{4.2}$$

By invoking the strong memoryless property, we have $\{V(y) - \tau_{j(y)+1} - \tau_{j(y)+2} - \dots - \tau_i - \tau_{i+1}: y \in A(Y_i) \cap A(Y_{i+1})\}$ is a set of independent, exponentially distributed random variables because $A(Y_i)$ is assumed to be a set of independent random variables. Since $\{y: j(y) = i + 1\}$ are assumed to be independent of the history of $\{X(t)\}$ to this point, we have established that $\{V(y) - \tau_{j(x)+1} - \tau_{(x)+2} - \dots - \tau_i - \tau_{i+1}: y \in A(Y_{i+1})\}$ is a set of independent, exponentially distributed random variables. •

Let Q be an $|\zeta| \times |\zeta|$ matrix given by

$$Q_{X,Y} = \begin{cases} \mu(x) & Y = X \cdot x \in \zeta \\ -\mu(X) & X = Y \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Theorem 6. $\{X(t), t \geq 0\}$ is a continuous time Markov chain with infinitesimal generator matrix Q.

Proof. Let $Y \in \zeta$, $|Y| = k$. By lemma 3, we know that, conditioned only on the greediness of Y,

$$W(Y \cdot y) - W(Y) = V(Y) - [\tau_{j(y)} + \tau_{j(y)+1} + \ldots + \tau_k]$$

for each $y \in A(Y)$. Thus, the implication of lemma 5 is that, conditioned on Y being greedy, the set $\{W(Y \cdot y) - W(Y): y \in A(Y)\}$ is a set of mutually independent, exponentially distributed random variables. •

Let $P_{X,Y}(t) = P[X(t) = Y \mid X(0) = X]$ for each pair $X, Y \in \zeta$.

Corollary 7. Let $Y \in \mathbf{B}$. Then $F(Y, t) = P[X(t) = Y] = P_{\varnothing,Y}(t)$

Proof. We designed $\{X(t)\}$ such that sample paths of this process are greedy access chains. Since (E, ζ, d^*) is assumed to be a greedy system, sample paths terminate in greedy optimal basic elements. •

Thus, we have shown that F(Y, t) is a first passage time distribution of a Markov process with state space ζ and generator matrix Q.

Example 6.

Reconsider the ongoing shortest path example. Suppose we wished to find F(abd, t). We can write the Kolmogorov equation $P'(t) = P(t)Q$ as a set of first order differential equations which may be solved iteratively. In the case of abd, the system is given by

$$P_{\varnothing,\varnothing}'(t) = -[\mu(a) + \mu(b)]\, P_{\varnothing,\varnothing}(t)$$

$$P_{\varnothing,b}'(t) = \mu(b)\, P_{\varnothing,\varnothing}(t) - [\mu(a) + \mu(e)]\, P_{\varnothing,b}(t)$$

$$P_{\varnothing,ba}'(t) = \mu(a) P_{\varnothing,b}(t) - [\mu(d) + \mu(e)]\, P_{\varnothing,ba}(t)$$

$$P_{\varnothing,bad}'(t) = \mu(d)\, P_{\varnothing,ba}(t).$$

When the rates are given by:

$\mu(a) = 2$; $\mu(b) = 1$; $\mu(c) = 4$;

$\mu(d) = 1$; $\mu(e) = 3$

this system has solution

$$F(bad, t) = P_{\varnothing,\, bad}(t) = P[W(bad) \leq W(Y) \text{ for all } Y \in \mathbf{B}, W(bad) \leq t]$$

$= 1/30 - 1/3\, e^{-3t} - 1/5\, e^{-5t} + 1/2\, e^{-4T}$. •

4.3 Performance Measures

In this section, we discuss some of the uses of the distribution $F(Y, t)$ in characterizing some measures of performance of the stochastic constant access system. While the fundamental equations

i) $F(P_i, t) = P[W(Y) \le t$ for some $Y \in \mathbf{B}$ such that $d^*(Y) = P_i]$

$= \Sigma_{Y:d^*(Y)=P_i} F(Y, t)$ for each $P_i \in \{d^*(Y): Y \in \mathbf{B}\}$

ii) $F(t) = P[W(Y) \le t$ for some $Y \in \mathbf{B}] = \Sigma_{Y\in \mathbf{B}} F(Y, t)$

iii) $P_Y = P[W(Y) \le W(Y')$ for all $Y' \in \mathbf{B}] = \lim_{t\to\infty} F(Y, t)$

are obviously valid, there exist more efficient methods for the computation of these distributions that do not require full knowledge of $F(Y, t)$ or even generation of every element of ζ. Historically, studies cited in the introduction were focused on the calculation of $F(t)$ for each example problem. Analysts undertaking analysis of some stochastic constant access system may have no need for the joint distribution.

Let $P = \{P_i: P_i = d^*(Y)$ for some $Y \in \mathbf{B}\}$. Let $\zeta_P = (\zeta - \mathbf{B}) \cup P$. Consider the modified continuous time Markov chain $\{X_P(t), t \ge 0\}$ on state space ζ_P with transition probability matrix Q^P given by

$Q^P_{X,Y} = Q_{X,Y}$ $\qquad$ $X, Y \in \zeta - \mathbf{B}$

$Q^P_{X,P_i} = Q_{X,Y}$ $\qquad$ $Y \in \mathbf{B}, d^*(Y) = P_i$.

Through this simple combining of basic elements of ζ, we have created a new process for which the absorption distribution is $P[X_P(t) = P_i \mid X_P(0) = \phi] = F(P_i, t)$. For the shortest path problem, $F(P_i, t)$ is the probability that P_i is the shortest path and it's length is less than or equal to t.

Arguably, the process $\{X_P(t)\}$ makes $\{X(t)\}$ obsolete, as we are rarely interested in all of the sample path information $\{X(t)\}$ can provide. More to the point, $\{X_P(t)\}$ seems to be very inefficient because it has several indistinguishable sample paths. The process $\{X_P(t)\}$ may be streamlined by performing the following procedure on the matrix Q^P and state space ζ_P. We call this procedure the **lumping procedure** for determining structure absorption.

initialize:

$L = P$.

while there exist $X, Z \in \zeta_P$ *such that* $Q^P_{X,W} = Q^P_{Z,W} = 0$ *for all* $W \in \zeta_P - L$ *and* $Q^P_{X,W} = Q^P_{Z,W}$ *for all* $W \in L$

for every $T \in \zeta_p$*, replace* Q^P_{TX} *with* $Q^P_{TX} + Q^P_{TZ}$

remove Z *from* ζ_P

replace L *with* $L \cup \{X \in \zeta_P\colon Q^P_{X,W} = 0$ *for all* $W \in \zeta_P - L\}$

repeat.

Note that performing the lumping procedure preserves the uppertriangularity of generator matrices. An analogous procedure may be constructed for any set of combined absorbing states, and the procedure extends to discrete time Markov chains in the obvious way. The theoretical importance of the lumping procedure is that it assures us that the system we choose to analyze has no redundant sample paths. In applications, we often use the lumping procedure in the abstract, lumping state space elements by some structural argument. In this case, the lumping procedure provides sufficient conditions for the validity of such an argument.

The distribution of the weight of the optimal basic element may be derived via a method similar to the one just described. Suppose we combined all of the basic elements into a single element Ψ, defining $\zeta_\Psi = (\zeta - B) \cup \{\Psi\}$ and $\{X_\Psi(t), t \geq 0\}$ with generator Q^Ψ defined by

$$Q^\Psi_{X,Y} = Q_{X,Y} \text{ if } Y \notin B$$

$$Q^\Psi_{X,\Psi} = \Sigma_{Y \in B}\ Q_{X,Y}.$$

Then $P[X_\Psi(t) = Y \mid X_\Psi(0) = \phi] = F(t)$ is the distribution of the weight of the minimum weight clutter element.

Example 7.

By lumping the states with identical determining structures and performing the lumping procedure, we greatly reduce the size of the state space. This reduces the sate space size from $|\zeta| = 14$ to $|\zeta_p| = 8$. If we are interested only in F(t), the distribution of the shortest path, we may further reduce the state space size to $|\zeta_\Psi| = 5$. On larger or denser networks, these reductions are more pronounced.

We need not generate the original state space ζ to generate ζ_p or ζ_Ψ. For the shortest path problem, nonbasic states with identical acessible sets may be lumped to give ζ_p. The same is true of $|\zeta_\Psi|$. Thus, we may perform the lumping procedure in the abstract, avoiding the computational work entailed in generating ζ.

4.4 Performance Measures Based on the Embedded Markov Chain

Let $\{Y_n, n \geq 0\}$ be the embedded discrete time Markov chain for $\{X(t)\}$. Thus, $\{Y_n\}$ has state space ζ and transition probability matrix P given by

$$P_{X,X \cdot x} = \mu(x)\,[\mu(X)]^{-1} \tag{4.4}$$

for $X \cdot x \in \zeta$. For the process $\{Y_n\}$, there is exactly one sample path from ϕ to each member of **B**. Thus, for $Y \in$ **B**, $|Y| = m$, we have the solution

$$P_Y = P[Y_G = Y] = \Pi_{i=1,2,\ldots,m}\ \mu(y_i)\,[\mu(Y_{i-1})]^{-1}. \tag{4.5}$$

Some measures easily derivable by summing different subsets of $\{P_Y: Y \in \mathbf{B}\}$ from $\{Y_n\}$ are

i) $P[d^*(Y_G) = P_i]$ for each $P_i \in P$

ii) $P[x \in d^*(Y_G)]$ for some $x \in E$

iii) $P[\,|Y_G| = m]$, $m = 1, 2, \ldots, |E|$

iv) $P[\,|d^*(Y)| = i]$, $i = 1, 2, \ldots, |E|$.

The first measure is the probability that a given determining structure is the minimum weight determining structure. For the shortest path example, this is the probability that a given path is shortest. The second density gives the probability that a given element of E is a member of the optimal determining structure, this is the probability that a given arc is on the shortest path. The third density function gives the distribution of the length of the optimal element. This is the number of steps taken by the greedy algorithm in solving the given problem--the distribution of the execution time of the algorithm. Finally, the last density gives the probability that the optimal determining structure is of a given length. This density is especially interesting if $\mu(x)$ is the same for each $x \in E$. Each of these measures may also be calculated efficiently by lumping absorbing states in the obvious way and performing the lumping procedure.

For any Markov chain $\{Z_n, n \geq 0\}$ with transition probability matrix P with strictly uppertriangular transition probability matrix, we solve the system

$P_\varnothing = 1;$

$P_Y = \Sigma_{X \in \zeta} P_X P_{X,Y}$

directly for the set $\{P_Y: Y \text{ is absorbing in } \{Z_n\}\}$.

Example 8.

Returning to the ongoing shortest path example, we have the following table for the basic elements:

$Y \in B$	$P[Y = Y_G]$	$Y \in B$	$P[Y = Y_G]$
ad	0.1111	acd	0.1111
be	0.2000	ace	0.3333
abd	0.0278	bad	0.0333
abe	0.0833	bae	0.1000

and for the directed (s, t) paths:

path set Pi	P[Pi is the shortest path in G]
{b, e}	0.3833
{a, d}	0.2833
{a, c, e}	0.3333

and criticality indices

arc x	P[x is on the shortest path in G]
a	0.6166
b	0.3833
c	0.3333
d	0.2833
e	0.7166

4.5 Expected Value Analysis

All of the Markov process's we have considered, regardless of any lumping performed, have uppertriangular generator matrices. We present here three simple formulae for the computation of the k^{th} moment of the conditional time until absorption of a Markov process with uppertriangular generator matrix. Let Q be the generator of the original process $\{X(t), t \geq 0\}$, let Q^{Ψ} and Q^{P} be the generators defined above. Let $Y \in \mathbf{B}$ and $P_i \in P$. Let U_X, $U_{X|P_i}$, and $U_{X|Y}$ be given by

$$U_X = \inf\{t: X_{\Psi}(t) \in \Psi \mid X_{\Psi}(0) = X\} \text{ for every } X \in \zeta_{\Psi}$$

$$U_{X|P_i} = \inf\{t: X_P(t) = P_i \mid X_P(0) = X, \lim_{t \to \infty} X_P(t) = P_i\} \text{ for every } X \in \zeta_P$$

$$U_{X|Y} = \inf\{t: X(t) = Y \mid X(0) = X, \lim_{t \to \infty} X(t) = Y\} \text{ for every } X \in \zeta$$

and define the moments $\tau_X(k)$, $\tau_{X|P_i}(k)$, and $\tau_{X|Y}(k)$ as $EU_X^{\,k}$, $EU_{X|P_i}^{\,k}$, and $EU_{X|Y}^{\,k}$, resp. We seek the quantities

$$\tau_\phi(k) = E[W(Y_G)^k]$$

$$\tau_{\phi\,|\,P_i}(k) = E[W(Y_G)^k \mid d^*(Y_G) = P_i]$$

$$\tau_{\phi\,|\,Y}(k) = E[W(Y_G)^k \mid Y_G = Y].$$

The first step equation for U_X is given by

$$U_X = [1 + \Sigma_{Z\in\zeta_\Psi} Q^\Psi{}_{X,Z} U_Z]\,[-Q^\Psi{}_{X,X}]^{-1}, \tag{4.6}$$

with $U_\Psi = 0$. Using moment generating functions, we derive

$$\tau_X(k) = [k\tau_X(k-1) + \Sigma_{Z\in\zeta_\Psi} Q^\Psi{}_{X,Z}\, \tau_Z(k)]\,[-Q^\Psi{}_{X,X}]^{-1} \tag{4.7}$$

for each $X \in \zeta_\Psi$ and each $k = 1, 2, \ldots$, and with $\tau_\Psi(k) = 0$. Using $\tau_X(0) = 1$ for all $X \in \zeta_\Psi$ as a boundary condition, we can directly compute $\{\tau_X(k)\colon X \in \zeta_\Psi\}$ from $\{\tau_X(k-1)\colon X \in \zeta_\Psi\}$ and Q^Ψ. In the case of $k = 1$, equation 4.7 simplifies to the familiar first step expected value equations

$$\tau_X(1) = [-Q^\Psi{}_{X,X}]^{-1} + \Sigma_{Z\in\zeta_\Psi} P^\Psi{}_{X,Z}\, \tau_Z(1) \tag{4.8}$$

where $P^\Psi{}_{X,Z} = Q^\Psi{}_{X,Z}\,[-Q^\Psi{}_{X,X}]^{-1}$ is the transition probability from X to Z for the process $\{X_\Psi(t)\}$.

With minor modifications to this methodology, we may find $\tau_{X\,|\,P_i}(k)$ for each $X \in \zeta_P$. Let us define the set $C_{P_i} = \{X \in \zeta_P\colon P[X_P(t) = P_i \mid X_P(0) = X] > 0\}$, thus C_{P_i} is the subset of ζ_P which communicates with P_i. Following 4.7, we find

$$\tau_{X\,|\,P_i}(k) = [k\tau_{X\,|\,P_i}(k-1) + \Sigma_{Z\in C_{P_i}} Q^P{}_{X,Z}\, \tau_{Z\,|\,P_i}(k)]\,[-Q^P{}_{X,X}]^{-1}, \tag{4.9}$$

$\tau_{P_i\,|\,P_i}(k) = 0$. Note that, consistent with a fundamental property of Markov process, the expected sojourn time in state X is unaffected by the conditioning. The form of 4.7 can also be used to find the expected value $\tau_{X\,|\,Y}(k)$,

$$\tau_{X\,|\,Y}(k) = [k\tau_{X\,|\,Y}(k-1) + \Sigma_{Z\subset Y} Q_{X,Z}\, \tau_{Z\,|\,P_i}(k)]\,[-Q_{X,X}]^{-1}. \tag{4.10}$$

As mentioned before, the original process $\{X(t), t \geq 0\}$ has a unique sample path to each basic element. Let $|Y| = m$, then for each $i < m$, 4.10 simplifies to

$$\tau_{Y_i\,|\,Y}(k) = k\tau_{Y_i\,|\,Y}(k-1)\,[\Sigma_{x\in A(Y_i)}\, \mu(x)]^{-1} + \tau_{Y_{i+1}\,|\,Y}(k-1).$$

Thus

$$\tau_{\phi\,|\,Y}(k) = k\Sigma_{i=0..m-1}[\Sigma_{x\in A(Y_i)}\, \mu(x)]^{-1}. \tag{4.11}$$

Example 9.

Revisiting our ongoing shortest path example, we calculate $\tau_0(1) = 0.6833$ and $\tau_0(2) = 0.6655$. This gives us a shortest (s, t) path length variance of 0.1986. Note that $E[v(a) + v(c) + v(e)] = 1.0833$, $E[v(a) + v(d)] = 1.500$, and $E[v(b) + v(e)] = 1.333$. Thus, if we were to naively estimate the length of the shortest path by the minimum of the expected value of the sums of arc lengths, we would overestimate the length by 0.4, over one standard deviation.

5.0 EXAMPLES FROM NETWORK OPTIMIZATION

In this final section, we present examples of constant access systems arising in network optimization. These constant access systems may be classified into two groups, those which have determining structures which are paths or cutsets in the network at hand, and those which involve the bottleneck objective function. Several of these constant access systems have already appeared in the literature as separate results. The purpose of presenting them here is to highlight their interrelationship and to build the reader's intuition with familiar structures. The k-trigger network example in Section 5.1 and all of the bottleneck objective function examples are presented in the literature for the first time in this paper. We shall illustrate the concepts and procedures described in sections 2 through 4 using these examples.

5.1 Stochastic Path and Cutset Optimization

In this section, we present some constant access systems in which the determining structures are paths or cutsets. The examples provided are the shortest path system (Kulkarni [1987]), the PERT system (Kulkarni and Adlakha[1986]), the maximum flow on directed planar networks (Kulkarni [1987]) and a new system which we call the k-trigger network.

Example 10. (Shortest Path Systems)

We have used an instance of the shortest path problem to motivate theory we have presented thus far. In this section, we present this problem in its full generality. Markov processes were first used to analyze shortest paths in Markov networks in Kulkarni [1987].

Let $G = (N, E)$ be a directed graph. Let s and t be two prespecified nodes in N, and suppose that there exists a (u, t) directed path for each $u \in N$. We will speak of a path as a set of directed arcs, the incident nodes being implicit. If $x \in E$ is given by the ordered pair (k, l), then we say head(x) = l and tail(x) = k. Let IN(n) denote the set $\{x \in E: \text{head}(x) = n\}$ and OUT(n) denote the set $\{x \in E: \text{tail}(x) = n\}$ for each $n \in N$.

Cast in the notation of constant access systems, Dijkstra's algorithm for finding the shortest path in G is given by the following algorithm.

initialize:

$X_0 = \emptyset$

$A(X_0) = OUT(s)$

$w(X_0) = 0$

for each $x \in E$, $r(x) = v(x)$

$i = 0$

While $|X_i \cap IN(t)| = \emptyset$

$x_{i+1} = argmin_{x \in A(X_i)}\ r(x)$

$X_{i+1} = X_i \cdot x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

$A(X_{i+1}) = A(X_i) - \{x_{i+1}\} - \{x \in E$: *no* $(tail(x), t)$ *directed path exists in* $(N, E - X_{i+1}) \cup OUT(head(x_{i+1}))\}$

$i = i + 1$

endwhile

Let ζ be the set of strings possibly generated by this algorithm for all nonnegative length functions. Thus, let X be a string of arcs, $|X| = i$ and let n = head(x_i). $X \in \zeta$ if and only if

i) (N, X) is an s-rooted directed tree;

ii) There exists at least one directed (n, t) path P such that none of the nodes implicit in P are incident with edges in X.

iii) Properties i and ii hold for every prefix of X.

From these properties, it is obvious that $X \in \mathbf{B}$ if and only if head(x_i) = t. The set $\{A(X): X \in \zeta\}$ is the set of uniformly directed cutsets (UDCs) in G along with the empty set.

The determining structure of each $Y \in \mathbf{B}$ is the unique directed (s, t) path contained in (N, Y). For $X \in \zeta - \mathbf{B}$, $d^*(X)$ is identified by

$$\Sigma_{d^*(X)} v(x) = \max_{Y \in \Gamma(X)} \Sigma_{d^*(X) \cap X} v(x),$$

$d^*(X)$ is the subset of maximum weight of an (s, t) path contained in X, where the maximization is performed over those (s, t) paths which are determining structures of basic elements for which X is a prefix.

The verification of the constant access property for (A, ζ, d^*) is a result of Sigal et. al. [1980]. They state that the intersection of a (s, t) path with a UDC is always a single element set. From our knowledge of Dijkstra's algorithm, we know that an arc x becomes accessible as soon as an arc incident with tail(x) is accessed. It remains accessible until it is accessed, or until there remain no $(\text{head}(x), t)$ paths which are not "crossed" by paths of accessed arcs. From this intuitive argument, we see that (A, ζ, d^*) has the interval property.

Because (A, ζ, d^*) is a greedy constant access system, we may apply the results from Section 3 to find $\{F(Y, t), Y \in \mathbf{B}, t \geq 0\}$ when G is a network with arcs with independent, exponentially distributed lengths. The shortest path system has the property that for each $Y \in \mathbf{B}$, (N, Y) contains a unique (s, t) path. If we were not interested in the actual order of access, we would lose nothing by lumping together strings with identical underlying sets--in fact, the lumping procedure would do this automatically. This reduction in the size of the state space is especially dramatic for dense graphs. If we were interested in only the length of the longest path, the lumping procedure would combine all states with the same accessible set. Thus, the state space could be replaced by the set of UDCs.

We may use the method for shortest path problems to analyze maximum flows in undirected planar networks with independent, exponentially distributed arc capacities by utilizing the topological dual of the network. It is well known that the minimum capacity cutset in a planar network G is identified as the shortest path in the dual network G', where the arc lengths in G' are the capacities in G.

Example 11. (Longest Path Systems)

Let $G = (N, E)$ be an acyclic directed network. Again let s and t be two prespecified nodes in N, and suppose that there exists a (u, t) path for each $u \in N$.. If we add $\{v(x): x \in A\}$, a set of weights representing durations of the tasks represented by the arcs, the resulting network is called a PERT network. In such a network, tail(x) = head(y) implies that activity y must be completed before activity x can commence, y has precedence over x. The goal of analysis of PERT networks is the identification of the longest path in G and its length. The well-known forward sweep algorithm for finding the longest path in G is given by the following

initialize:

$X_0 = \varnothing$

$A(X_0) = OUT(s)$

$w(X_0) = 0$

for each $x \in E$, $r(x) = v(x)$

$i = 0$

While $|Xi - E| \neq \emptyset$

$x_{i+1} = argmin_{x \in A(X_i)}\ r(x)$

$X_{i+1} = X_i \cdot x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

if $|X_i \cap IN(head(x_{i+1}))| = indegree(head(x_{i+1}))$ *then*

$A(X_{i+1}) = A(X_i) \cup OUT(head(x_{i+1}))$

else

$A(X_{i+1}) = A(X_i)$

$i = i + 1$

endwhile

Let ζ be the set of strings possible generated by nonnegative length functions and let **B** be the set of basic elements of ζ. This structure is the well-known schedule greedoid, so–named by Korte' and Lovasz [1984]. The determining structure of $Y \in \mathbf{B}$, $|Y| = m$, is constructed as follows:

i) $y_m \in d^*(Y)$;

ii) for each $y \in d^*(Y)$, identify $\{y_{i_1}, y_{i_2}, ..., y_{i_k}$: head(yij) = tail(y)$\}$. Let $i^* = \max_{j=1..k}\{i_j\}$, then $y_i{}^* \in d^*(Y)$.

Thus, the last arc accessed which points into tail(y_m) is in $d^*(Y)$, the last arc accessed pointing into that arc is in $d^*(Y)$, and so on. This backward chain ends when an arc y with tail(y) = s is identified as a member of $d^*(Y)$, forming a (s, t) path.

Let us verify that (A, ζ, d^*) has the constant access property. Consider $X \in \zeta - \mathbf{B}$. A(X) is obviously a subset of an (s, t) UDC. Again using the result of Sigal et. al. [1980],.we know A(X) contains at most one arc from each (s, t) path. We need to show that this number is exactly one for (s, t) paths in $\Gamma(X)$.

Let $Y \in \mathbf{B}$ and consider $k < |Y|$. Let $i = \max\{j \leq k: y_j \in d^*(Y)\}$, thus y_i is the last element of $d^*(Y)$ accessed at stage k. Let $y \in d^*(Y)$ with tail(y) = head(y_i). Because $\{Y\} = A$, and by the definition of i, $y \in A(Y_i)$. Thus, $A(Y_i)$ contains exactly one element of $d^*(Y)$.

The interval property on (A, ζ) is straightforward to show, the proof is provided by Korte' and Lovasz [1984], who called all greedoids with the interval property "alternative precedence structures," or APS greedoids. The greedy algorithm performed on (A, ζ, d^*) is the obvious forward sweep algorithm, see Lawler [1976].

Example 12. (Maximum Flow on Planar Directed Graphs)

Let G = (N, E) be a directed planar graph and let nodes s and t be prespecified source and sink nodes. We assume that for each $u \in N$, there is at least one (u, t) path in G. Let $v:E \rightarrow \Re^+$ be a function giving the capacity values for each of the arcs.

Let {C1, C2, ..., Ck} be the set of UDCs in G. The maximum (s, t) flow MF is given by

$$MF = \Sigma_{x \in C_i} \; v(x).$$

The constant access system constructed must have the property that $\{d^*(X), X \in \zeta\} = \{C_1, C_2, ..., C_k\}$. The development of the greedy constant access scheme to find the maximum feasible flow in G, given by Itai and Shiloach [1979], depends on the existence of the topmost path TP(E') in the subgraph G' = (N, E') for any subset $E' \subset E$ such that an (s, t) path in G'. The algorithm is given as follows:

initialize:

$X_0 = \varnothing$

$w(X_0) = 0$

for each $x \in E$, $r(x) = v(x)$

$i = 0$

While $TP(E - X_i) \neq \varnothing$

$x_{i+1} = argminx \in TP(E - X_i)\, v(x)$

$X_{i+1} = X_i . x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

for each $x \in TP(E - Xi)$

$r(x) = r(x) - r(x_{i+1})$

$i = i + 1$

endwhile

Let ζ be the set of strings which this algorithm might generate for all possible choices of nonnegative capacity functions v, and let **B** be the set of basic elements. Let $Y_n \in$ **B**. There is no topmost path in $G' = (N, E - Y_n)$, but $y_n \in TP(E - Y_{n-1})$. Thus Y_n must contain an (s, t) UDC, and this UDC must contain y_n. Chaining backwards, we are able to identify the UDC $d^*(Y_n)$ by lemma 1.

This algorithm, known as the path filling algorithm, is shown to produce the maximum flow $w(Y_G) = MF$ in Itai and Shiloach. $\{A(X): X \in \zeta\}$ is the set of topmost paths in G. The constant access property holds as a result $|C_i \cap P_j| = 1$ for each path P_j in the set $\{P_1, P_2, ..., P_m\}$ of (s, t) paths.

Let us verify the interval property for (E, ζ). For each $x \in E$ we know that once $x \in A(X_i)$, x is on the current topmost path, and will remain in the set of accessible arcs until either x is accessed or the topmost path no longer includes x. Let k be the smallest index such that $x \notin A(X_k)$. This implies that X_k contains at least one arc of every (s, t) path which involves x. In stage n, n>k, x cannot be on the topmost path of $(N, E - X_n)$ because $X_k \subset X_n$. Thus, (E, ζ) has the interval property. Recapping, the system (E, ζ, d^*) has the interval and constant access properties, is a greedy system, and $\{d^*(Y): Y \in \mathbf{B}\}$ is a clutter on E. Hence (E, ζ, d^*) is a constant access system.

Let us consider the case where the capacities on the arcs are a set of independent, exponentially distributed random variables. The full process on the state space ζ will give us the absorption time distributions $F(Y, t) = P[X(t) = Y] = P[Y_G = Y, w(Y) \le t]$, for each $Y \in \mathbf{B}$. By lumping the members of Y by their determining structures, we may find $F(C_i, t)$, the probability that C_i is the minimum capacity UDC in G and its capacity is less than or equal to t. This may be accomplished by lumping strings with the same accessible sets, reducing the state space size to the number of (s, t) paths.

The probability that the capacity of a given arc constrains the maximum flow of the network, the criticality of the arc, is seen to depend on the flow value. If we wish to find the criticality of a set of arcs K, we suggest a lumping of the basic states into two sets. One set contains all basic states for which at least one member of K is a member of the determining structure, and is lumped into the single state Ψ_1. The other set contains the complement of the first and is labeled Ψ_2. The lumping procedure should then be executed on the remaining state space, as suggested in Section 4.3.

If one were interested in reducing the capacity of the network to t_0 or less by interdicting arcs in the set K, one would be interested in $1 - P[X(t_0) = \Psi_1] =$ P[K contains at least one critical arc and flow is greater than t_0] as a measure of the importance of the set K at the flow value t_0. Similarly, we can construct the embedded Markov chain as given in Section 4.4 to give the probability that a string is a minimum weight string, the probability that a UDC is the minimum capacity UDC, and the probability that a set of arcs K contains critical arcs.

Example 13. (K-trigger Networks)

Consider a directed acyclic network G = (N, E) with arc length function $v: E \to \Re^+$. In this problem formulation, it is intuitive to regard the arcs of E as activities as in PERT formulations. Suppose that each node $n \in N$ has a requirement constant kn so that activities emanating from n cannot commence until k_n activities pointing into n have completed. Suppose we are given prespecified nodes s and t. Assume that indegree(s) = 0. We are interested in the time of completion of the k_t^{th} arc in IN(t) to complete.

For the PERT formulation, k_n = indegree(n) for each $n \in N$, no activity in OUT(n) can start until all activities in IN(n) are complete. Thus, the time when the last arc completes is the length of the longest path in G. If $k_n = 1$ for all $n \in N$, the time of completion of the first arc in IN(t) is the length of the shortest path in G. Thus, This problem which we name the k-trigger network problem generalizes both shortest and longest path problems on directed graphs.

The k-trigger network model is also useful in modelling the assembly of information products such as tracking information assembled by an air control system. Each of the activities represents some data collection task or some ambiguity resolution activity. Each activity must have several independent inputs which it uses to produce a coherent product or conduct some data collection requiring an initial solution. These activities may also represent the command task, giving permission for subsequent activities to begin.

We may find the completion time of the k_t^{th} activity in IN(t) by the following greedy algorithm

initialize:

$X_0 = \varnothing$

$w(X_0) = 0$

for all $x \in E$, $r(x) = v(x)$

$A(X_0) = OUT(s)$

$i = 0$

While $|X_i \cup IN(t)| < k_t$

$x_{i+1} = argmin_{x \in A(X_i)}\ r(x)$

$X_{i+1} = X_i \cdot x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

for all $x \in A(X_i)$

$r(x) = r(x) - r(x_{i+1})$

if $|X_{i+1} \cup IN(head(x_i))| = k_{head(x_i)}$ *then*

$A(X_{i+1}) = A(X_i) - IN(head(x_i)) \cup OUT(head(x_i)) - \{x \in E$: *no* $(tail(x), t)$ *path exists in* $(N, E - X_{i+1}) \cup OUT(head(x_{i+1}))\}$

else

$A(X_{i+1}) = A(X_i) - \{x_{i+1}\} - \{x \in E$: *no* $(tail(x), t)$ *path exists in* $(N, E - X_{i+1}) \cup OUT(head(x_{i+1}))\}$

$i = i + 1$

endwhile

Let ζ be the set of strings generated over all length functions v and let B be the set of basic strings. Clearly, if this algorithm generates the basic string Y_n, we know that y_n is the $k_t{}^{th}$ arc in IN(t) to complete, thus clearly $y_n \in d^*(Y_n)$. Furthermore, recall that $j(y_n)$ is the minimum j such that $y_n \in A(Y_j)$. From the algorithm, we see that $head(y_{j(y_n)}) = tail(y_n)$, and $y_{j(y_n)}$ is the $k_{tail(y_n)}{}^{th}$ arc to be accessed in IN(tail(y_n)). Hence, it must be on the longest path from s to tail(y_n) in the subgraph $(N, \bar{Y}_n)$. Thus, we can chain backwards to identify $d^*(Y_n)$ as the longest (s, t) path in the subgraph (N, Y_n). The length of this path is the shortest completion time of the assembly process.

Because $d^*(Y_n)$ is always an (s, t) path, we have that $\{d^*(Y): Y \in B\}$ is a clutter on E. The interval property holds because an arc x becomes accessible when $k_{tail(x)}$ arcs in IN(tail(x)) have been accessed and remains accessible until it is accessed or until $k_{head(x)}$ arcs in IN(head(x)) are accessed. In the latter event, x remains inaccessible thereafter.

The constant access property is more difficult to verify. One must recognize that given that X_i is generated by the above algorithm, $A(X_i)$ is a partial UDC. One must show that any (s, t) path P_1 for which $|A(X_i) \cap P_1| = 0$ is dominated in length by at least one path P_2 for which $|A(X_i) \cap P_2| = 1$. P_1 will never be the longest (s, t) path in (N, Y) for any $Y \supset X_i$, $Y \in B$. Hence, $P_1 \notin \Gamma(X_i)$. Thus, the constant access property holds for this system. The system has the three properties necessary for it to be a constant access system.

We can determine the probability that a given arc is critical, a member of $d^*(Y_G)$, by using the embedded Markov chain. We can derive interesting performance measures such as the distribution of the total time of assembly, the probability that a given arc is critical given the assembly time is less than t_0, and the probability that a set of arcs contains a critical arc given that the assembly time exceeds t_0 by using the lumping procedure to alter the matrix Q.

5.2 Systems with Bottleneck Objective Functions

In this section, we discuss three constant access systems which use the bottleneck objective function. The bottleneck objective function, first specified by Edmonds and Fulkerson [1970], is used to find the structure with the lowest cost where the cost of a structure is the weight of the maximum weight element in the structure. In this section, we present examples where the structures are cutsets, paths and trees. These are given in network reliability, bottleneck shortest path, and bottleneck minimum weight tree.

Example 14. (Network Reliability)

Consider a network G = (N, E) with undirected arcs whose usability is perishable. For instance, consider a communications network with links which fail with time. The network contains a set S of sender nodes and a disjoint set T of terminals. We are interested in the time it takes for the network to fail to the point where there is no path of usable arcs from some member of S to some member of T, that is, the time that S and T are no longer completely connected. Note that this problem has (s, t) and (s, K) reliability as special cases.

Let $\{C_1, C_2, ..., C_k\} = C$ be the set of (S, T) cutsets, and let $v:E \rightarrow \Re^+$ be a function giving the lifelength of each arc. The quantity of interest is the time of failure TF of the longest lived arc in the shortest lived cutset,

$$TF = \min_{C_i \in C} \max_{x \in C_i} v(x).$$

TF is a well-known instance of the bottleneck objective function on the set C. The obvious algorithm for finding TF is given as

initialize:

$X_0 = \varnothing$

$w(X_0) = 0$

for all $x \in E$, $r(x) = v(x)$

$i = 0$

While S and T are completely connected in $(N, E - X_i)$

$x_{i+1} = \mathrm{argmin}_{x \in E - X_i}\, r(x)$

$X_{i+1} = X_i \cdot x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

for all $x \in E - X_i$

$r(x) = r(x) - r(x_{i+1})$

$i = i + 1$

endwhile

At each stage, $w(X_i) = v(x_i)$, thus if the algorithm terminates in n steps, xn is the longest lived arc in the shortest lived cutset, and $w(X_n) = v(x_n)$ is the time of disconnection. Let ζ be the set of strings generated by this algorithm over all nonnegative failure time functions.

The system (E, ζ, d^*) has determining structures which are singleton sets, $d^*(X_i) = \{x_i\}$. Thus $\{d^*(Y): Y \in \mathbf{B}\} = E$, thus $\{d^*(Y): Y \in \mathbf{B}\}$ is a clutter on the set E. For each $X_i \in \zeta$,

$$\Gamma(X_i) = \{d^*(Y): X_i \subset Y\} = E - X_i = A(X_i),$$

thus the constant access property is satisfied, $|d^*(Y_n) \cup A(X_i)| = |y_n| = 1$ for each Y_n such that $X_i \subset Y_n$. The interval property holds because every arc is accessible until it is accessed or the algorithm terminates. Thus, (E, ζ, d^*) is a constant access system.

Clearly, the network reliability problem for the case where the lifelengths are independent exponentials is solvable using Markov processes. By appropriate lumping of basic elements, we may determine for each time of failure the probability that a given cutset is longest lived and the probability that a given arc is the arc which determines the lifelength of the system. The embedded Markov chain may be used to find these quantities without regard for failure time.

Example 15. (Bottleneck Shortest Path)

We now turn attention to a system which finds the minimum weight path where path weights are given by the longest arc in the path. Let $\{P_1, P_2, ..., P_n\} = P$ be the set of (s, t) paths in G. The bottleneck (s, t) path problem may be stated as: find the (s, t) path with the shortest longest arc. That is, find Pi which minimizes

$$BP = \min_{P_i \in P} \max_{x \in P_i} v(x).$$

The following greedy algorithm finds the minimizing path and the optimal value BP.

initialize:

$X_0 = \varnothing$

$w(X_0) = 0$

for all $x \in E$, $r(x) = v(x)$

$i = 0$

While X_i *does not contain an (s, t) path*

$x_{i+1} = argmin_{x \in E-X_i}\ r(x)$

$X_{i+1} = X_i \cdot x_{i+1}$

$w(X_{i+1}) = w(X_i) + r(x_{i+1})$

for all $x \in E - X_i$

$r(x) = r(x) - r(x_{i+1})$

$i = i + 1$

endwhile.

BP is given by $w(Y_G)$. Let ζ be the set of all strings generated by this algorithm. As with the reliability problem, the interval property holds because each arc is accessible at each stage prior to termination. The determining structure of each $Y_m \in B$ is y_n, and $A(X_i) = E - X_i$ for all nonbasic X_i. Hence the constant access property must hold. Obviously $\{d^*(Y): Y \in B\} = E$, a clutter on itself.

Thus, (E, ζ, d^*) is a constant access system. The methodology in sections 4.3 through 4.5 may be used to find the distribution of BP, the moments of BP, and the probability that a given arc is the arc determining BP. Note that the minimizing path is not unique, and every (s, t) path in (N, X_i) is equally qualified to be the minimum weight path.

Example 16. (Bottleneck Spanning Tree)

Let $\{T_1, T_2, ..., T_n\}$ be the set of arc sets of spanning trees in the undirected network G = (N, E). The bottleneck spanning tree problem, very similar to the bottleneck path problem, is to find the spanning tree with the shortest longest arc,

$$BST = \min_{T_i \in T} \max_{x \in T_i} v(x).$$

The greedy algorithm to solve the bottleneck spanning tree problem is identical to the bottleneck path problem, except that we terminate the algorithm when Xi contains a spanning tree. The determining structures are singleton arc sets, and the interval and constant access properties hold. Hence, the system (E, ζ, d^*) is a constant access system.

6.0 CONCLUSION

In this work, we have constructed a new structural framework for randomly weighted network optimization called a constant access system. If a structure has the properties we have given here, we may find the joint distribution of the minimum weight basis element and its weight by calculating the absorption time distribution of a Markov process. We provide the construction of this Markov process, and demonstrate expansively how we may exploit the structure of the Markov process to derive measures of stochastic performance for the general problem.

We have established that the mechanism of transition in a Markov process is to greedily minimize the sojourn time in every state. This fortunate property allowed us to model the execution path of the greedy algorithm as the sample path in a Markov process.

APPENDIX A.

In this appendix we present the proof to lemma 2.

Lemma 2. Let (E, ζ, d^*) be a greedy constant access system. Let X be a greedily generated string, $|X| = n$. Then for each $x \in A(X)$, $x \in d^*(X \cdot x)$.

Proof. For each i, define the following two subsets of $\Gamma(Y \cdot y)$. Let $\mathbf{D} = \{J \in \Gamma(Y \cdot y)$ such that

$$\Sigma_{z \in J \cap X \cdot x}\ v(z) = \max_{J \in \Gamma(X \cdot x)} \Sigma_{z \in J \cap X \cdot x}\ v(z)$$

Thus, $d^*(X \cdot x) = J \cap X \cdot x$ for any $J \in \mathbf{D}$. Let $\mathbf{K} = \{J \in \Gamma(X \cdot x): x \in J\}$. Let $\mathbf{D_k} = \{J \in \Gamma(X_k)$ such that

$$\Sigma_{z \in J \cap X_k}\ v(z) = \max_{J \in \Gamma(X_k)} \Sigma_{z \in J \cap X_k}\ v(z)$$

Thus, $d^*(X_k) = J \cap X_k$ for any $J \in \mathbf{D_k}$. Let $\mathbf{K_k} = \{J \in \Gamma(X_k): x \in J\}$ We will show that if $\mathbf{K_k} \subset \mathbf{D_k}$ for k = 0, 1, ..., i, then $\mathbf{K} \subset \mathbf{D}$. This will prove the lemma by induction.

Let i = 0. In this case the proposition holds trivially since $\mathbf{K} = \mathbf{D} = \{d^*(Z): Z \in \mathbf{B}\}$. Thus, suppose that for k = 0, 1, ..., i, $\mathbf{K_k} \subset \mathbf{D_k}$. In order to show that this implies that $\mathbf{K} \subset \mathbf{D}$, we need to treat two cases: j(x) = i, and j(x) < i.

<u>Case 1</u>: j(x) = i, thus x is accessible only after x_i has been accessed. Let $K \in \mathbf{K}$. Then lemma 1 gives us that $x_i \in K$ so $K \in \mathbf{K_i}$. Thus, by induction hypothesis, $K \in \mathbf{D_i}$. We must show that $K \in \mathbf{D}$.

Let $J \in \mathbf{D}$. Then

$$\Sigma_{z \in J \cap X \cdot x}\ v(z) \le \Sigma_{z \in K \cap X \cdot x}\ v(z) + v(x_{i+1})$$

$$= \Sigma_{z \in K \cap X \cdot x}\ v(z). \tag{A.1}$$

Thus, $K \in \mathbf{D}$.

<u>Case 2</u>: j(x) = h < i. Let $K \in \mathbf{K}$. Let $J \in \mathbf{D}$ and assume that $K \notin \mathbf{D}$. Thus

$$\Sigma_{z \in K \cap X \cdot x}\ v(z) < \Sigma_{z \in J \cap X \cdot x}\ v(z). \tag{A.2}$$

The constant access property guarantees that $x_i \notin K$, and the induction hypothesis guarantees that $x_i \in J_i$. Hence

$$\Sigma_{z \in K \cap X_{i-1}}\ v(z) + v(x) = \Sigma_{z \in K \cap X \cdot x}\ v(z)$$

$$< \Sigma_{z \in J \cap X_{i+1}}\ v(z) = \Sigma_{z \in J \cap X_{i-1}}\ v(z) + v(x_i), \tag{A.3}$$

the inequality resulting from equation A.2. Note that $\{x_i, x\} \subset A(X_{i-1})$. It is straightforward to show that if inequality A.3 holds, then $G(X_{i-1}) \neq x_i$. In fact, $w(X_{i-1} \cdot x_i) > w(X_{i-1} \cdot x)$, and both x_i and x were accessible at stage $i - 1$. This shows that for case 2, if we assume $K \in \mathbf{K}$ and $K \notin \mathbf{D}$, we can derive a contradiction.

Thus, we have shown for both cases that $\mathbf{K} \subset \mathbf{D}$. The lemma follows directly. •

ACKNOWLEDGEMENT

This work was sponsored by the Naval Postgraduate School Research Foundation.

REFERENCES

[1] COREA, G. AND V. G. KULKARNI. 1987. Minimum Cost Routing on Stochastic Networks. Technical Report UNC/OR/TR-87/8, University of North Carolina, Chapel Hill, NC.

[2] EDMONDS, J. AND D. R. FULKERSON. 1970. Bottleneck Extrema. *Journal of Combinatorial Theory* 8, 299-306.

[3] KORTE' B. AND L. LOVASZ. 1984. Greedoids - A Structural Framework for the Greedy Algorithm. *Progress in Combinatorial Optimization.* Academic Press, Canada.

[4] KULKARNI, V. G. 1987. Shortest Paths in Networks with Exponentially Distributed Arc Lengths. Networks 16, 255-274.

[5] KULKARNI, V. G. AND V. G. ADLAKHA. 1986. Markov and Markov-Regenerative PERT Networks. *Operations Research* 34, 769-781.

[6] KULKARNI, V. G. AND V. G. ADLAKHA. 1987. Maximum Flow in Planar Networks with Exponentially Distributed Arc Capacities. Stochastic Models I, 263-289.

[7] ITAI, A. AND Y. SHINLOACH. 1979. Maximum flow in planar networks. SIAM J. Comput. 8, 135-150

[8] LAWLER, E. L. 1976. *Combinatorial Optimization: Networks and Matroids.* Holt, Reinhart, and Winston, New York.

[9] NEUTS, M. 1981. *Matrix-Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, MD.

[10] SCHASSBERGER, R. 1976. On the Equilibrium Distribution of a Class of Finite-State Generalized Semi-Markov Processes. *Mathematics of Operations Research* 1, 395-406.

[11] SIGAL, C. E., A. A. B. PRITSKER, AND J. J. SOLBERG. 1980. The Use of Cutsets in Monte Carlo Analysis of Stochastic Networks. *Mathematics and Computers in Simulation*, **21**, p. 376-384.

[12] WEISS, G. 1986. Stochastic Bounds on Distributions of Optimal Value Functions with Applications to pert, Network Flows and Reliability. *Operations Research* **34**, 595-605.

DISTRIBUTION LIST

	NO. OF COPIES
Library (Code 0142) Naval Postgraduate School Monterey, CA 93943-5000	2
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Office of Research Administration (Code 012) Naval Postgraduate School Monterey, CA 93943-5000	1
Center for Naval Analyses 4401 Ford Avenue Alexandria, VA 22302-0268	1
Library (Code 55) Naval Postgraduate School Monterey, CA 93943-5000	1
Operations Research Center, Rm E40-164 Massachusetts Institute of Technology Attn: R. C. Larson and J. F. Shapiro Cambridge, MA 02139	1
Koh Peng Kong OA Branch, DSO Ministry of Defense Blk 29 Middlesex Road SINGAPORE 1024	1
Institute for Defense Analysis 1800 North Beauregard Alexandria, VA 22311	1
Arthur P. Hurter, Jr. Professor and Chairman Dept. of Industrial Engineering and Management Sciences Northwestern University Evanston, IL 60201-9990	1
Professor H. G. Daellenbach Dept. of Operations Research University of Canterubry Christchurch, NEW ZEALAND	1

Department of Operations Research, Code 55 Naval Postgraduate School Monterey, CA 93943-5000	5
Michael Page Bailey, Code 55Ba Naval Postgraduate School Monterey, CA 93943-5000	20
V. G. Kulkarni Department of Operations Research The University of North Carolina at Chapel Hill Chapel Hill, NC 27599	1
J. George Shantikuman The Management Science Group School of Business Administration University of California at Berkeley Berkeley, CA 93720	1